11-16-00 A

PTO/SB/05 (12/97)

Please type a plus sign (+) inside this box → [+]

Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No	TI-28977
First Named Inventor or Application Identifier	Sanjive Agarwala, et al.
Title	Request Queue Manager in Transfer Controller With Hub and Ports
Express Mail Label No.	EL547741984

APPLICATION ELEMENTS

See MPEP Chapter 600 concerning utility patent application contents

ADDRESS TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231

1. [X] *Fee Transmittal Form *(e.g., PTO/SB/17)* *(Submit an original, and a duplicate for fee processing)*
2. [X] Specification *[Total Pages* [19] *]* *(preferred arrangement set forth below)*
   - Descriptive title of the Invention
   - Cross References to Related Applications
   - Statement Regarding Fed sponsored R&D
   - Reference to Microfiche Appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
3. [X] Drawing(s) *(35 USC d113)* *[Total Sheets* [3] *]*
4. Oath or Declaration *[Total Pages* [2] *]*
   - a. [X] Newly Executed (original or copy)
   - b. [ ] Copy from a prior application (37 CFR §1.63(d)) *(for continuation/divisional with Box 17 completed)*

   [Note Box 5 below]

   - i. [ ] DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR §1.63(d)(2) and 1 33(b)
5. [ ] Incorporation By Reference *(useable if Box 4b is checked)* The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein
6. [ ] Microfiche Computer Program *(Appendix)*
7. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   - a. [ ] Computer Readable Copy
   - b. [ ] Paper Copy (identical to computer copy)
   - c. [ ] Statement verifying identical of above copies

ACCOMPANYING APPLICATION PARTS

8. [X] Assignment Papers (cover sheet & Documents(s))
9. [ ] 37 CFR §3.73(b) Statement *(when there is an assignee)* [X] Power of Attorney
10. [ ] English Translation Document *(if applicable)*
11. [ ] Information Disclosure Statement (IDS)/PTO-1449 [ ] Copies of IDS Citations
12. [X] Preliminary Amendment
13. [X] Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
14. [ ] *Small Entity Statement(s) *(PTO/SB/09-12)* [ ] Statement filed in prior application Status still proper and desired
15. [ ] Certified Copy of Priority Document(s) *if foreign priority is claimed)*
16. [ ] Other.

*A new statement is required to be entitled to pay small entity fees, except where one has been filed in a prior application and is being relied upon

17. **If a CONTINUING APPLICATION,** *check appropriate box and supply the requisite information below and in a preliminary amendment:*

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: / .

Prior application information: ***Examiner*** _____ ***Group / Art Unit:*** _____

18. CORRESPONDENCE ADDRESS

[X] *Customer Number or Bar Code Label* **23494** *or Correspondence address below*

(Insert Customer No. or Attach bar code label here)

NAME	**Robert D. Marshall, Jr.**		
ADDRESS			
CITY	*STATE*	*ZIP CODE*	
COUNTRY	*TELEPHONE* **972-917-5290**	*FAX*	**972-917-4418**

Name (Print/Type)	**Robert D. Marshall, Jr.**	*Registration No. (Attorney/Agent)*	**28,527**
Signature	Robert D Marshall Jr	*Date* **November 15, 2000**	

Burden Hour Statement: This form is estimated to take 0 2 hours to complete Time will vary depending upon the needs of the individual case Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office Washington, DC 20231 DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS SEND TO Assistant Commissioner for Patents Box Patent Application, Washington, DC 20231

PTO/SB/17 (1/98)
Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

FEE TRANSMITTAL

Patent fees are subject to annual revision on October 1
These are the fees effective October 1, 1997
Small Entity payments must be supported by a small entity statement, otherwise large entity fees must be paid. See Forms PTO/SB/09-12.

Express Mailing Label No.: **EL547741984**

TOTAL AMOUNT OF PAYMENT ($) 710.00

Complete If Known	
Application Number	
Filing Date	November 15, 2000
First Named Inventor	Sanjive Agarwala, et al.
Examiner Name	
Group / Art Unit	
Attorney Docket No.	TI-28977

METHOD OF PAYMENT

1. ☒ The Commissioner is hereby authorized to charge to the following Deposit Account,

Deposit Account Number: 20-0668

Deposit Account Name: Texas Instruments Incorporated

☒ Charge any additional fee required or credit any overpayment

☐ Charge all indicated fees and any additional fee required or credit any overpayment

2. ☐ **Payment Enclosed:**

☐ Check ☐ Money Order ☐ Other

FEE CALCULATION

1. BASIC FILING FEE

Large Fee Code	Entity Fee ($)	Small Fee Code	Entity Fee ($)	Fee Description	Fee Paid
101	710	201	395	Utility filing fee	$ 710
106	310	206	165	Design filing fee	$
107	480	207	270	Plant filing fee	$
108	760	208	395	Reissue filing fee	$
114	150	214	75	Provisional filing fee	$

SUBTOTAL (1) ($) 710

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	9	-20**=	0	x	18	=	0
Independent Claims	1	-3** =	1	x	80	=	0
Multiple Dependent					260	=	

**or number previously paid, if greater; For Reissue, see below

Large Fee Code	Entity Fee ($)	Small Fee Code	Entity Fee ($)	Fee Description
103	18	203	11	Claims in excess of 20
102	78	202	41	Independent Claims in excess of 3
104	260	204	135	Multiple dependent claims in excess of 3
109	78	209	41	**Reissue independent claims over original patent
110	18	210	11	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 0

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Fee Code	Entity Fee ($)	Small Fee Code	Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	380	216	200	Extension of time within second month	
117	870	217	475	Extension of time within third month	
118	1,360	218	755	Extension of time within fourth month	
128	1,850	228	1,030	Extension of time within fifth month	
119	300	219	155	Notice of Appeal	
120	300	220	155	Filing a brief in support of an appeal	
121	260	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a pubic use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,210	241	660	Petition to revive - unintentional	
142	1,210	242	660	Utility issue fee (or reissue)	
143	430	243	225	Design issue fee	
144	580	244	335	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (time number of properties)	
146	760	246	395	Filing a submission after final rejection (37 CFR 1.129(a))	
149	760	249	395	For each additional invention to be examined (37 CFR 1.129(b))	

Other fee (specify) _____

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3)

SUBMITTED BY		Complete (if applicable)	
Typed or Printed Name	Robert D. Marshall, Jr.	Reg Number	28,527
Signature	Robert D. Marshall, Jr.	Date: November 15, 2000	Deposit Account User ID

Burden Hour Statement This form is estimated to take 0.2 hours to complete Time will vary depending upon the needs of the individual case Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231 DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS SEND TO Assistant Commissioner for Patents, Washington, DC 20231

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of: TI-28977

Sanjive Agarwala, et al.

Serial No:

Filed: November 15, 2000

For: Request Queue Manager in Transfer Controller With Hub and Ports

PRELIMINARY AMENDMENT

"EXPRESS MAILING" Mailing Label No. **EL547741984**. Date of Deposit: **November 15, 2000**. I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 CFR 1.10 on the date shown above and is addressed to: Ass't Commissioner for Patents, Washington, D.C. 20231.

Robin E. Barnum

Ass't Commissioner for Patents
Washington, DC 20231

Dear Sir:

Please amend the specification by inserting before the first line, the following sentence:

--This application claims priority under 35 USC §119(e)(1) of Provisional Application Number 60/169,451, filed December 7, 1999.--

Respectfully submitted,

Robert D. Marshall, Jr.
Attorney for Applicants
Reg. No. 28,527

Texas Instruments Incorporated
P.O. Box 655474, MS 3999
Dallas, TX 75265
(972) 917-5290

REQUEST QUEUE MANAGER IN TRANSFER CONTROLLER WITH HUB AND PORTS

Sanjive Agarwala
Iain Robertson
David A. Comisky
Charles L. Fuoco
Christopher L. Mobley

TECHNICAL FIELD OF THE INVENTION

The technical field of this invention is digital signal processing and more particularly control of data transfers within a digital signal processing system.

BACKGROUND OF THE INVENTION

Digital signal processing (DSP) differs significantly from general purpose processing performed by micro-controllers and microprocessors. One key difference is the strict requirement for real time data processing. For example, in a modem application, it is absolutely required that every sample be processed. Even losing a single data point might cause a digital signal processor application to fail. While processing data samples may still take on the model of tasking and block processing common to general purpose processing, the actual data movement within a digital signal processor system must adhere to the strict real-time requirements of the system.

As a consequence, digital signal processor systems are highly reliant on an integrated and efficient direct memory access (DMA) engine. The direct memory access controller is responsible for processing transfer requests from peripherals and the digital signal processor itself in real time. All data movement by the direct memory access must be capable of occurring without central processing unit (CPU) intervention in order to meet the real time requirements of the system. That is, because the CPU may operate in a software tasking model where scheduling of a task is not as tightly controlled as the data streams the tasks operate on require, the direct memory access engine must sustain the burden of meeting all real time data stream requirements in the system.

The early direct memory access has evolved into several successive versions of centralized transfer controllers and more recently into the transfer controller with hub and ports architecture. The transfer controller with hub and ports architecture is described in U.K. Patent Application No. 9901996.9 filed April 10, 1999 entitled "TRANSFER CONTROLLER WITH HUB AND PORTS ARCHITECTURE."

A first transfer controller module was developed for the TMS330C80 digital signal processor from Texas Instruments. The transfer controller consolidated the direct memory access function of a conventional controller along with the address generation logic required for servicing cache and long distance data transfer, also called direct external access, from four digital signal processors and a single RISC (reduced instruction set computer) processor.

The transfer controller architecture of the TMS330C80 is fundamentally different from a direct memory access in that only a single set of address generation and parameter registers is required. Prior direct memory access units required multiple sets for multiple channels. The single set of registers, however, can be utilized by all direct memory access requestors. Direct memory access requests are posted to the transfer controller via set of encoded inputs at the periphery of the device. Additionally, each of the digital signal processors can submit requests to the transfer controller. The external encoded inputs are called "externally initiated packet transfers" (XPTs). The digital signal processor initiated transfers are referred to as "packet transfers" (PTs). The RISC processor could also submit packet transfer requests to the transfer controller.

The transfer controller with hub and ports introduced several new ideas concepts. The first was uniform pipelining. New digital signal processor devices containing a transfer controller with hub and ports architecture have multiple external ports, all of which look identical to the hub. Thus peripherals and memory may be freely interchanged without affecting the hub. The second new idea is the concept of concurrent execution of transfers. That is, up to N transfers may occur in parallel on the multiple ports of the device, where N is the number of channels in the transfer controller with hub and ports core. Each channel in the transfer controller with hub and ports core is functionally just a set of registers. This set of registers tracks the current source and destination addresses, the word counts and other parameters for the transfer. Each channel is identical, and thus the number of channels supported by the transfer controller with hub and ports is highly scaleable.

Finally the transfer controller with hub and ports includes a mechanism for queuing transfers up in a dedicated queue memory. The TMS320C80 transfer controller permitted only was one transfer outstanding per processor at a time. Through the queue memory provided by the transfer controller with hub and ports, processors may issue numerous transfer requests up to the queue memory size before stalling the digital signal processor.

SUMMARY OF THE INVENTION

In a transfer controller with hub and ports, the request queue manager is the hub front-end which operates in conjunction with a transfer request bus to handle incoming requests. The queue manager control unit sorts these requests by their respective priorities and stores them in the queue manager memory. The queue manager memory could take several forms, but is preferably RAM based.

The queue manager memory is partitioned into multiple queues. Each queue represents a priority level with which access requests in that queue should be performed by the transfer controller hub. The number of channels in the request queue manager is highly scalable, and controlled via a set of queue bounds registers within the request queue manager. The queue manager also processes read requests from the transfer controller hub and supplies the next queued request to the hub on a per channel basis.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other aspects of this invention are illustrated in the drawings, in which:

Figure 1 illustrates in a functional block diagram, the basic principal features of the transfer controller with hub and ports;

Figure 2 illustrates the request queue manager interface to the transfer controller with hub and ports hub unit; and

Figure 3 illustrates the functional block diagram of the request queue manager.

DETAILED DESCRIPTION OF THE INVENTION

The request queue manager function is a crucial part of the centralized transfer controller with hub and ports architecture. To understand its various performance aspects it is helpful to consider first the transfer controller as a whole.

The transfer controller with hub and ports transfer controller with hub and ports architecture is optimized for efficient passage of data throughout a digital signal processor chip. Figure 1 illustrates a block diagram of the principal features of the transfer controller with hub and ports. It consists of a system of a single hub 100 and multiple ports 111 through 115. At the heart of the hub is the transfer controller with hub and ports hub control unit 109 which acts upon request and status information to direct the overall actions of the transfer controller.

The transfer controller with hub and ports functions in conjunction with a transfer request bus having a set of nodes 117, which bring in transfer request packets at input 103. These transfer request bus nodes individually receive transfer requests packets from transfer requestors 116 which are processor-memory nodes or other on-chip functions which send and receive data.

Secondly, the transfer controller uses an additional bus, the data transfer bus having a set of nodes 118, to read or write the actual data at the requestor nodes 116. The data transfer bus carries commands, write data and read data from a special internal memory port 115 and returns read data to the transfer controller hub via the data router 150 at inputs 104.

The transfer controller has, at its front-end portion, a request queue manager 101 receiving transfer requests in the form of transfer request packets at its input 103. Request queue manager 101 prioritizes, stores and dispatches these as required.

Request queue manager 101 connects within the transfer controller hub unit 100 to the channel request registers 120 which receive the data transfer request packets and process them. In this process, request queue manager 101 first prioritizes the transfer request packets and assigns them to one of the N channel request registers 120. Each of the N channel request registers 120 represents a priority level.

If there is no channel available for direct processing of the transfer request packet, it is stored in the queue manager memory 102. Queue manager memory 102 is preferably a random access memory (RAM). The transfer request packet is then assigned at a later time when a channel becomes available. The channel registers interface with the source 130 and destination 140 control pipelines which effectively are address calculation units for source (read) and destination (write) operations.

Outputs from these pipelines are broadcast to M ports through the transfer controller ports I/O subsystem 110. I/O subsystem 110 includes a set of hub interface units, which drive the M possible external ports units. Four such external ports are shown in Figure 1 as external ports 111 through 114. The external ports units (also referred to as application units) are clocked either at the main processor clock frequency or at a different external device clock frequency. The external device clock frequency may be lower than or

higher than the main processor clock frequency. If a port operates at its own frequency, synchronization to the core clock is required.

As an example of read-write operations at the ports, consider a read from external port node 112 followed by a write to external port node 114. First the source pipeline addresses port 112 for a read. The data is returned to the transfer controller hub through the data router unit 150. On a later cycle the destination control pipeline addresses port 114 and writes the data at port 114. External ports as described here do not initiate transfer requests but merely participate in reads and writes requested elsewhere on the chip. Read and write operations involving the processor-memory (transfer requestors) nodes 116 are initiated as transfer request packets on the transfer request bus 117. The queue manager 101 processes these as described above. On a later cycle a source pipeline output (read command/address) is generated which is passed at the internal memory port to the data transfer bus 118 in the form of a read. This command proceeds from one node to the next in pipeline fashion on the data transfer bus. When the processor node addressed is reached, the read request causes the processor-memory node to place the read data on the bus for return to the data router 150. On a later cycle, a destination pipeline output passes the corresponding write command and data to the internal memory port and on to the data transfer bus for writing at the addressed processor node.

The channel parameter registers 105 and port parameters registers 106 hold all the necessary parametric data as well as status information for the transfer controller hub

pipelines to process the given transfer. Both pipelines share some of the stored information. Other portions relate specifically to one pipeline or the other.

Figure 2 illustrates the interface of request queue manager 101 to the transfer controller hub unit boundary and particularly the request queue manager communications with the channel request registers 200, channel parameters registers 105 and port parameters registers 106. Channel parameters registers 105 and port parameters registers 106 store critical data regarding for example, types of transfers, mode information, status, and much other information critical to the transfer process.

Channel request registers 200 pass information used in the source control pipeline 130 for generation of the read/ pre-write commands 251. Similarly, channel request registers 200 pass information used in the destination control pipeline 140 for the generation of write command/write data words 252. Read response data 104 from the ports is returned to the destination pipeline via the data router unit 150.

Detailed View of the Queue Manager

Request queue manager 101, illustrated in Figure 3, operates in conjunction with the transfer request bus 212 and handles incoming requests. Queue manager control unit 201 sorts these requests by their priorities and stores them in the queue manager memory 102. Queue manager memory 102 may take several forms, but is preferably random access memory (RAM) based.

Queue manager memory 102 may be partitioned into multiple queues. Each such queue represents a priority level with

which access requests in that queue should be performed by the transfer controller hub. Figure 3 shows six priority levels 210, labeled CH0 through CH5. The number of channels in request queue manager 101 is highly scalable, and controlled via a set of queue bounds registers 203 within request queue manager 101. The number of queue bounds registers determines the number of queues. Each queue within request queue manager 101 may be of different size, however each queue must contain at least enough memory space to store a single transfer request.

Request queue manager 101 also processes channel requests from the transfer controller hub 211 and supplies the next queued request to the hub on a per channel (1 channel per priority level) basis.

Request Queue Manager-Transfer Request (TR) Bus Interface

Transfer requests from the transfer request bus 212 sent to request queue manager 101 are sorted and stored in the queue manager memory 102 until the transfer controller hub can process them. A priority field of the first double word on an incoming transfer request determines its priority and is used to select into which of the queues the transfer request will be placed.

Write pointer logic, which is part of the queue bounds registers 203 and read/write pointer block 204, will increment the address only after each write and then only if the signal allowing queue manager memory 102 bypass is not active. If this increment causes the address to be equal to the next successive queue bounds value, then the next address will be reset to the original queue bounds value for that queue.

As an optional enhancement, it is possible to bypass request queue manager 101 when an incoming transfer request packet is addressed to a currently empty queue and the transfer controller hub channel to which that queue corresponds is idle. This condition is detected by comparing the read and write pointers for the appropriate queue. If the read and write pointers are equal, the queue channel is empty. Further inspection of the channel status within transfer controller hub 100 can detect the idle condition. Thus the transfer request parameters may be directly loaded into the transfer controller hub channel registers 120. This has the advantage of reducing latency on access requests. In the case of queue manager storage bypass, request queue manager 101 read and write pointers 204 for the appropriate channel are not incremented.

Request Queue Manager Hub Interface

Transfer controller hub 100 accepts the next entry in request queue manager 101 as soon as the respective channel becomes available (i.e. the previous request for that channel has completed). These reads are on a per-channel basis with the highest priority being serviced first. This requires a "data ready" signal 221 to be sent back to each of the channels indicating when the respective channel is being serviced by request queue manager 101. Data is sent to the transfer controller hub via the path 222 shown in Figure 3. For each read from queue manager memory 102, the read pointer 204 is incremented except in the case of queue manager storage bypass as noted above.

Request queue manager 101 can issue only one store or one transfer request at a time. The "Channel Request from Hub" signal 211 has a priority level higher than the "Request from the TR Bus" signal 212. If there is a conflict between channel request from hub 211 and request from transfer request bus 212, the latter is interrupted until the channel request is served. Channel request service is allocated the CH0 priority level, which is the highest priority level, and is served first.

Request Queue Manager Requestor Disable

Requestor disable register block 206 is included in request queue manager 101. Transfer request "reads" that have a requestor ID that has been disabled via requestor disable register block 206 will be ignored. Request queue manager 101 will continue to advance to the next request in the queue until one that is not disabled is located, or the queue is empty. Request disable register block 206 is memory mapped and can be accessed through internal memory port 115 illustrated in Figures 1 and 2.

Request Queue Manager Requestor ID Acknowledgement (QACK)

Request queue manager 101 has an extra duty to inform the transfer request nodes 116 when one of the transfer requests has been taken out of its priority queue and passed to the transfer controller hub for processing. This is required to notify the individual transfer request nodes 116 when reserved queue slots have freed up. Then transfer request nodes 116 can increment their counter value and issue another request if the counter was zero. The requestor ID QACK signal 226 is

sent when it is read out of queue manager memory 102 for the transfer controller hub. The requestor ID QACK signal 226 is passed along transfer request nodes 116 on another bus paralleling the transfer request bus. Each transfer request node 116 can verify that the requestor ID QACK was sent during that cycle and update their counters if applicable. A valid bit is included on this additional bus to qualify when the requestor ID was read out.

Request Queue Manager Datapath Summary

In summary, the datapath in request queue manager 101 consists of three main parts. The first is the queue manager control unit 201. This functional block takes in the channel request transfer request packet 211, transfer request node transfer request packet 212, and the queue manager "queue empty" 218 information from the hub. It then decodes the channel register transfer request packet, compares it with the "queue empty" information to check if any transfer request packet is ready to be transferred. If there is a transfer packet and a "channel ready", queue manager control unit 201 then issues the channel number, issues a read enable to request queue manager 101 main data path to read the packets from queue manager RAM. If there is no transfer packet or "channel ready", queue manager control unit 201 will then do a transfer request write to queue manager memory 102 provided there is a transfer request. If there is a transfer request and a "channel ready", queue manager control unit 201 then issues the channel number and read enable to request queue manager 101 main data path to read the packets from queue

manager memory 102 and also do the transfer request write to queue manager memory 102.

The second part of request queue manager 101 datapath is the main datapath. It has N read pointer counters and N write pointer counters for N queues, all a part of block 203/204. For this example N equals six because there are six priority levels. Based on whether a read or write operation is issued, request queue manager 101 datapath selects the read or write pointer of the corresponding queue. The datapath also detects whether the pointer will pass beyond the queue boundary. If the next address will go beyond the upper queue boundary, it will reset the next address to the lower queue boundary.

Request queue manager 101 main datapath also has N-1 queue bound registers, which may be globally memory mapped. These N-1 queue bound registers are also a part of queue bounds registers 203. The queue bounds registers define the lower and upper bounds of the N queues. When they are reset to a new value, both the corresponding read and write pointers are reset to the bottom of the queue immediately.

WHAT IS CLAIMED IS:

1. A method of prioritizing data transfer requests serviced by a centralized data transfer unit comprising the steps of:

receiving transfer request packets, each transfer request packet indicating a desired data transfer and a priority level within a hierarchy of a plurality of priority levels;

storing each received transfer request packet in a queue memory, each received transfer request packet stored in a first-in-first-out fashion within each priority level;

detecting availability of free data transfer channel among a plurality of data transfer channels within said centralized data transfer processor; and

dispatching a next transfer request packet in said first-in-first-out fashion to the corresponding free data transfer channel.

2. The method of claim 1, wherein:

said step of storing each received transfer request packet in a queue memory includes

storing each received transfer request packet in a random access memory fashion,

defining an address range within said queue memory allocated to each priority level with a corresponding queue bounds register,

storing a next input transfer request packet for a priority level at address within said queue memory indicated by a corresponding queue write pointer indicating address location, and

reading a next output transfer request packet for a priority level from an address within said queue memory indicated by a corresponding queue read pointer.

3. The method of claim 2, further comprising the steps of:

incrementing the corresponding queue write pointer upon storage of a transfer request packet in queue memory; memory and

decrementing the corresponding queue read pointer upon transfer of a transfer request packet to a data transfer channel.

4. The method of claim 2, further comprising the steps of:

dynamically defining said address range allocated to a priority level by dynamically writing to a corresponding queue bounds register.

5. The method of claim 1, further comprising the steps of:

detecting if a channel corresponding to a priority level within the queue is empty; and

if a received transfer request packet of has a priority level detected to be empty

bypassing storing said transfer request packet in said queue memory, and

dispatching said transfer request packet directly to the corresponding free data transfer channel.

6. The method of claim 1, further comprising the steps of:

generating said transfer request packets at each of a plurality of transfer request nodes; and

upon dispatching of a transfer request packet to a free data transfer channel, sending a queue acknowledge signal to said transfer request node originating said transfer request packet.

7. The method of claim 1, further comprising the steps of:

generating said transfer request packets at each of a plurality of transfer request nodes, said transfer request packet identifying said originating transfer request node;

disabling dispatching transfer request packets originating from selected transfer request nodes; and

ignoring transfer request packets originating from disabled transfer request nodes.

8. The method of claim 8, wherein:

said step of disabling dispatching transfer request packets originating from selected transfer request nodes includes writing at least one to a corresponding location within a request disable register.

9. The method of claim 9, wherein:

the number of data transfer channels equals the number of priority levels.

ABSTRACT

A transfer controller with hub and ports is viewed as a communication hub between the various locations of a global memory map. A request queue manager serves as a crucial part of the transfer controller. The request queue manager receives these data transfer request packets from plural transfer requests nodes. The request queue manager sorts transfer request packets by their priority level and stores them in the queue manager memory. The request queue manager processes dispatches transfer request packets to a free data channel based upon priority level and first-in-first-out within priority level.

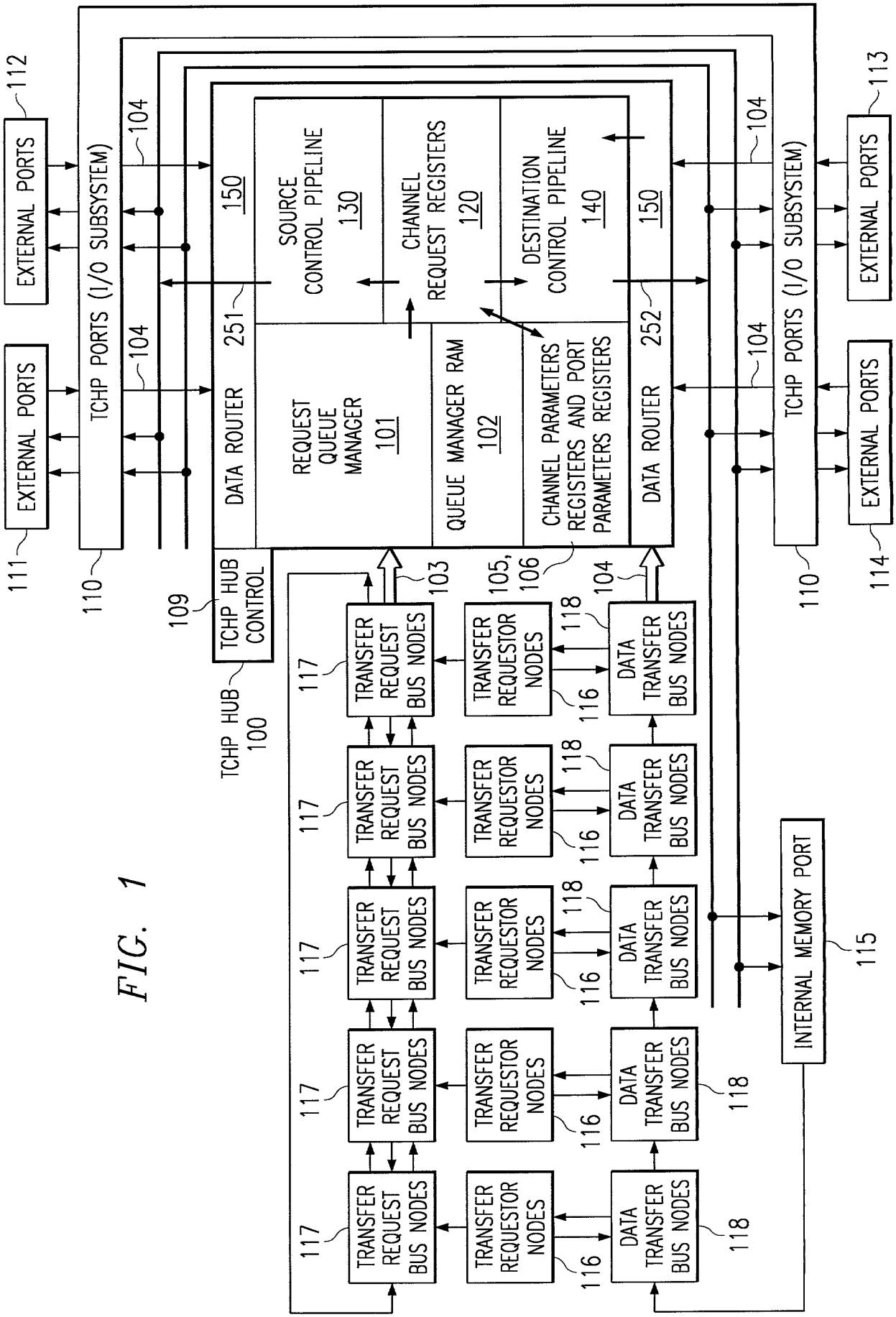
TCHP HUB 100
TCHP HUB CONTROL
109
DATA ROUTER 251
REQUEST QUEUE MANAGER 101
SOURCE CONTROL PIPELINE 130
CHANNEL REQUEST REGISTERS 120
DESTINATION CONTROL PIPELINE 140
QUEUE MANAGER RAM 102
CHANNEL PARAMETERS REGISTERS AND PORT PARAMETERS REGISTERS
DATA ROUTER 252
150
103
105, 106
104
TCHP PORTS (I/O SUBSYSTEM)
110
EXTERNAL PORTS
111
112
113
114
TRANSFER REQUEST BUS NODES
TRANSFER REQUESTOR NODES
DATA TRANSFER BUS NODES
116
117
118
INTERNAL MEMORY PORT
115

FIG. 1

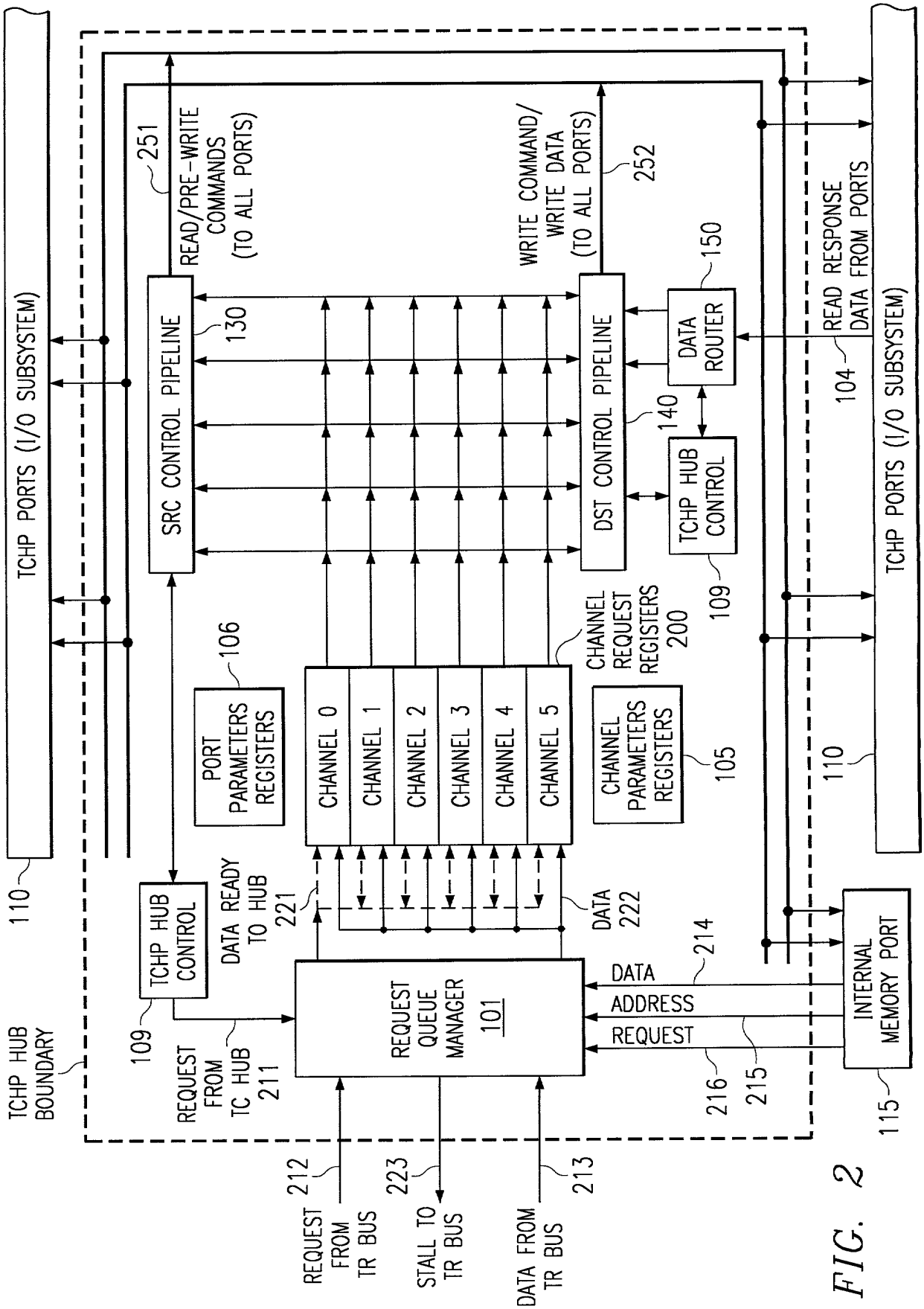
TCHP PORTS (I/O SUBSYSTEM)
110
TCHP HUB BOUNDARY
SRC CONTROL PIPELINE
130
READ/PRE-WRITE COMMANDS (TO ALL PORTS)
251
WRITE COMMAND/ WRITE DATA (TO ALL PORTS)
252
DST CONTROL PIPELINE
140
DATA ROUTER
150
TCHP HUB CONTROL
109
READ RESPONSE DATA FROM PORTS
104
TCHP PORTS (I/O SUBSYSTEM)
110
PORT PARAMETERS REGISTERS
106
CHANNEL 0
CHANNEL 1
CHANNEL 2
CHANNEL 3
CHANNEL 4
CHANNEL 5
CHANNEL REQUEST REGISTERS 200
CHANNEL PARAMETERS REGISTERS
105
TCHP HUB CONTROL
109
DATA READY TO HUB
221
DATA
222
REQUEST QUEUE MANAGER
101
DATA
ADDRESS
REQUEST
214
216
215
INTERNAL MEMORY PORT
115
REQUEST FROM TC HUB
211
REQUEST FROM TR BUS
212
STALL TO TR BUS
223
DATA FROM TR BUS
213

FIG. 2

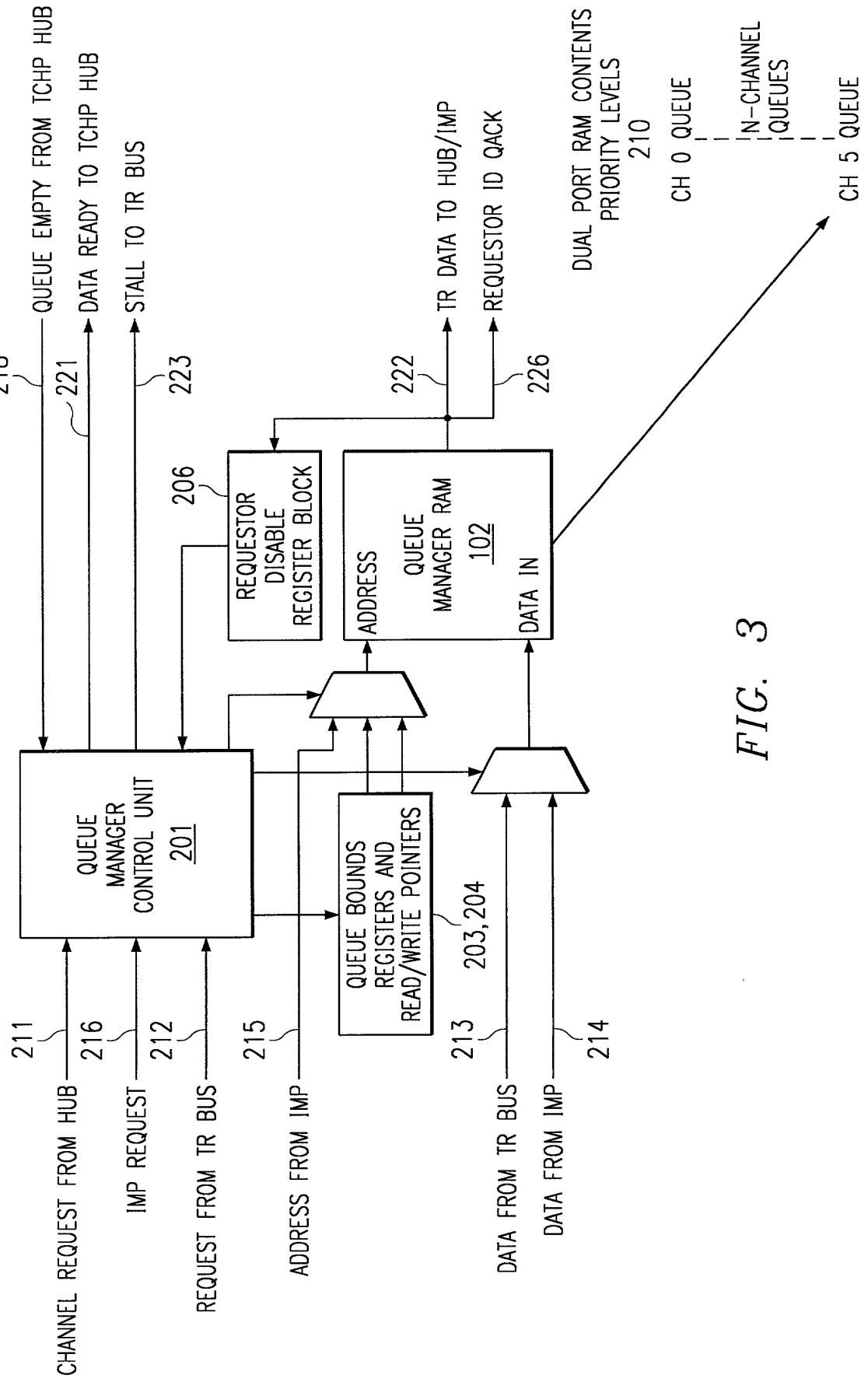
QUEUE EMPTY FROM TCHP HUB
DATA READY TO TCHP HUB
STALL TO TR BUS
TR DATA TO HUB/IMP
REQUESTOR ID QACK
DUAL PORT RAM CONTENTS
PRIORITY LEVELS
210
CH 0 QUEUE
N-CHANNEL
QUEUES
CH 5 QUEUE
218
221
223
222
226
206
REQUESTOR
DISABLE
REGISTER BLOCK
ADDRESS
QUEUE
MANAGER RAM
102
DATA IN
QUEUE
MANAGER
CONTROL UNIT
201
QUEUE BOUNDS
REGISTERS AND
READ/WRITE POINTERS
203, 204
211
216
212
215
213
214
CHANNEL REQUEST FROM HUB
IMP REQUEST
REQUEST FROM TR BUS
ADDRESS FROM IMP
DATA FROM TR BUS
DATA FROM IMP

FIG. 3

ATTORNEY'S DOCKET NO.
TI-28977

APPLICATION FOR UNITED STATES PATENT
DECLARATION AND POWER OF ATTORNEY

As a below named inventor, I declare that my residence, post office address and citizenship are as stated below next to my name; that I verily believe that I am the original, first and sole inventor if only one name is listed below, or an original, first and joint inventor if plural inventors are named below, of the subject matter which is claimed and for which a patent is sought on the invention entitled as set forth below, which is described in the attached specification **of Application Serial No. 60/169,451, filed 12/07/99**; that I have reviewed and understand the contents of the specification, including the claims, as amended by any amendment specifically referred to in the oath or declaration; that no application for patent or inventor's certificate on this invention has been filed by me or my legal representatives or assigns in any country foreign to the United States of America; and that I acknowledge my duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, Section 1.56;

I further declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

TITLE OF INVENTION:

Request Queue Manager in Transfer Controller with Hub and Ports

POWER OF ATTORNEY: I HEREBY APPOINT THE FOLLOWING ATTORNEYS TO PROSECUTE THIS APPLICATION AND TRANSACT ALL BUSINESS IN THE PATENT AND TRADEMARK OFFICE CONNECTED THEREWITH

Frederic J. Telecky, Jr., #29,979; Jay M. Cantor, #19,906; C. Alan McClure, #31,041; William B. Kempler, #28,228; Robert D. Marshall, #28,527; Gerald E. Laws, 39,268; James F. Hollander, #27,802; Robby Holland, #33,304; Rebecca Mapstone-Lake, #36,208.

SEND CORRESPONDENCE TO:	DIRECT TELEPHONE CALLS TO:
Robert D. Marshall, Jr. Texas Instruments Incorporated P.O. Box 655474, MS 3999 Dallas, TX 75265	Robert D. Marshall, Jr. (972) 917-5290

NAME OF INVENTOR: (1)	NAME OF INVENTOR: (2)	NAME OF INVENTOR: (3)
Sanjive Agarwala	Iain Robertson	David A. Comisky
RESIDENCE & POST OFFICE ADDRESS:	RESIDENCE & POST OFFICE ADDRESS:	RESIDENCE & POST OFFICE ADDRESS:
2608 Foxcreek Drive Richardson, Texas 75082	10, Grange Lane Bedfordshire MK44 3TT	812 Wynnpage Lane Plano, Texas 75075
COUNTRY OF CITIZENSHIP:	COUNTRY OF CITIZENSHIP:	COUNTRY OF CITIZENSHIP:
India	United Kingdom	United States
SIGNATURE OF INVENTOR: x Sanjive Agarwala	SIGNATURE OF INVENTOR:	SIGNATURE OF INVENTOR: x
DATE: x 02/22/00	DATE: 22 FEB 2000	DATE: x 02/22/00

ATTORNEY'S DOCKET NO.
TI-28977

NAME OF INVENTOR: (4)	NAME OF INVENTOR: (5)	NAME OF INVENTOR: (6)
Charles L. Fuoco	Christopher L. Mobley	N/A
RESIDENCE & POST OFFICE ADDRESS:	RESIDENCE & POST OFFICE ADDRESS:	RESIDENCE & POST OFFICE ADDRESS:
401 Misty Meadow Drive Allen, Texas 75013	216 Whispering Hills Drive Coppell, Texas 75019	
COUNTRY OF CITIZENSHIP:	COUNTRY OF CITIZENSHIP:	COUNTRY OF CITIZENSHIP:
United States	United States	
SIGNATURE OF INVENTOR: x	SIGNATURE OF INVENTOR: x	SIGNATURE OF INVENTOR:
DATE: x 2/22/2000	DATE: x 2/22/00	DATE: